Fast entropy clustering of sparse high dimensional binary data

Marek Śmieja
Szymon Nakoneczny
Jacek Tabor
Faculty of Mathematics and Computer Science
Jagiellonian University
Lojasiewicza 6, 30-348 Kraków, Poland
Email: {marek.smieja, szymon.nakoneczny, jacek.tabor}@ii.uj.edu.pl

***Abstract*—We introduce Sparse Entropy Clustering (SEC) which uses minimum entropy criterion to split high dimensional binary vectors into groups. The idea is based on the analogy between clustering and data compression: every group is reflected by a single encoder which provides its optimal compression. Following the Minimum Description Length Principle the clustering criterion function includes the cost of encoding the elements within clusters as well as the cost of clusters identification. Proposed model is adopted to the sparse structure of data – instead of encoding all coordinates, only non-zero ones are remembered which significantly reduces the computational cost of data processing. Our theoretical and experimental analysis proves that SEC works well with imbalance data, minimizes the average entropy within clusters and is able to select the correct number of clusters.**

I. Introduction

Clustering is usually the first choice for analysis large amount of high dimensional data when no prior knowledge is given [1]. It is an unsupervised technique which discovers meaningful groups based only on the internal structure of data and assumed similarity criterion. Nevertheless, many typical clustering approaches applied to high dimensional sparse problems, which occur often, e.g., in cheminformatics [2] or natural language processing (NLP) [3], fail due to the curse of dimensionality which makes the distance between objects to be less informative [4]. Moreover, their straightforward use could lead to substantial increase of computational cost.

In this paper we introduce Sparse Entropy Clustering (SEC) for grouping sparse binary vectors contained in high dimensional space. This is an information-theoretic approach (based on the entropy criterion function) which follows the analogy between clustering and data compression. Its basic idea states that data should be split into such clusters which could be efficiently compressed by using separate encoder for each group. Consequently, **the elements with similar probability distributions are grouped together**. Following the Minimum Description Length Principle (MDLP) [5], [6] our method also takes into account the complexity of the model by introducing the cost of clusters (encoders) identification (see Figure 2). This regularization term **allows to select the optimal number of groups**, see Theorem 3.1 and Figure 4 for details of our theoretical analysis.

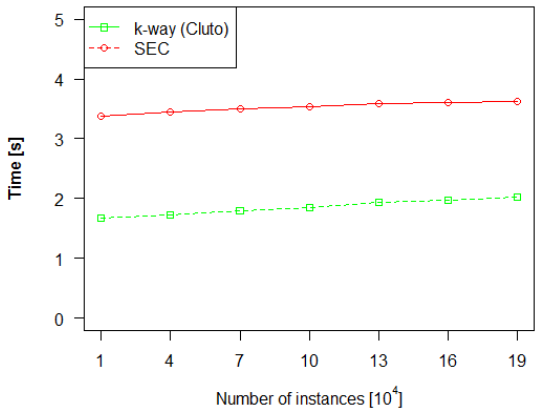

Fig. 1. Running times of a single iteration of SEC and k-way for clustering of artificial data set with a constant number of non-zero bits in the entire set. Both algorithms are optimized for processing sparse high dimensional data.

SEC adjusts the optimal compression model to the sparse structure of data, i.e., instead of encoding all coordinates, only non-zero positions are remembered (see Figure 3). This substantially reduces the statistical code-length for describing data and the corresponding computational cost for its processing (see Figure 1). The constructed clustering criterion function, reflecting the cost of data compression by applying multi-encoders model, can be practically optimized in the k-means-like style using fast iterative Hartigan procedure [7]. **The running time of a single iteration is determined by the total number of non-zero bits in the entire data set.** The influence of the number of instances and data dimension is marginal (see Figure 1).

In the experimental study we demonstrate that our method is able to discover true structure (probability distribution) of data even for highly imbalanced sets (see Figure 6 and 7). Its verification on texts clustering shows that SEC gives significantly higher compatibility with reference partition for selected examples than baseline k-medoids and hierarchical clustering techniques as well as k-way method implemented in Cluto software (see Figure 8). In the case of real data set of chemical compounds its performance is comparable to k-way and better than the baseline (see Figure 10). Moreover,

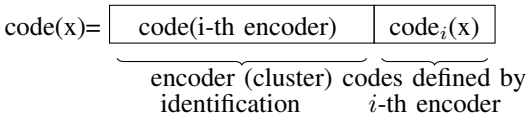

Fig. 2. Multi-encoders model.

SEC has a tendency to produce partitions with a relatively low average entropy within clusters (see Figures 9 and 11) – this criterion is often used in NLP.

The paper is organized as follows. A brief summary of related work is presented in the next section. The third section introduces SEC clustering model and contains our main theoretical result concerning model simplification. Practical realization of the algorithm is given in fourth section. The experiments are included in fifth section, while the sixth part contains a conclusion.

II. Related work

Our method follows an information-theoretic approach and is related with minimum entropy clustering introduced in [8], [9] and further developed in [10], [11]. Contrary to the classical entropy clustering which focuses on minimizing the entropy within groups, SEC also includes the cost of maintaining the model. This partially refers to MDLP [5] – different realizations of this idea in the case of clustering were considered in [12], [13]. However, in our setting we do not take into account the entire complexity of the model, but only its most meaningful part defining the cost of clusters identification similarly to the cross-entropy clustering (CEC) approach introduced in [6]. This regularization provides a natural way to discover the optimal number of groups.

Most of entropy-based clustering methods were defined for processing dense, either continuous or discrete data structures [6], [11], [13]. The proposed model is adopted to work efficiently with sparse binary vectors. This kind of data appears very often in real life examples as text mining [3] or computer-aided drug design [2]. Since binary attributes can be seen as categories, our method can be compared not only with techniques focusing on binary data type, but also with a wide range of categorical data clustering techniques [14]. In particular, one of the most powerful softwares for clustering high dimensional sets is Cluto [15] which pays a particular attention to the sparsity of data.

Due to the categorical nature of binary data, the standard euclidean distance is not an optimal choice for comparing objects. A brief summary of available similarity measures for binary (and categorical) vectors is presented in [12], [16]. It is worth to mention that typical clustering methods which rely on pairwise similarity, as hierarchical approaches [17] or spectral techniques [18], [19], can be directly used for this type of data by providing an appropriate distance matrix. On the other hand, to apply the popular k-means method [20] we have to be able to calculate the mean of a cluster. Since the center is not-well defined in non-euclidean space, the corresponding k-medoids method [21] or Wards approach [22], [23] are more frequently used.

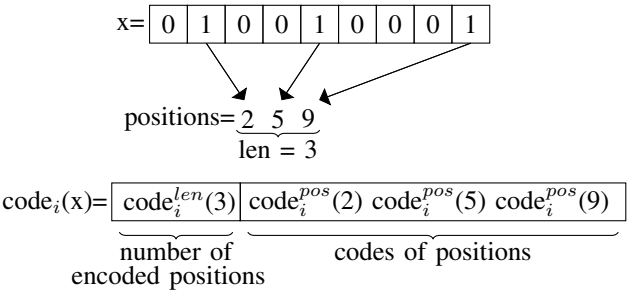

Fig. 3. Sparse data coding.

III. Theoretical model

Minimum Description Length Principle (MDLP) states that the best description of data is the one that leads to the best compression rate. In this section we will present our theoretical model of clustering high dimensional sparse binary vectors which follows the concept of MDLP. We start with discussing a basic sparse compression model which uses a single encoder for the entire data set. The next part covers its extension to the case of k-encoders which allows to split data into groups. We also consider simplification of the model which allows to select optimal number of clusters (see Theorem 3.1).

A. Sparse coding model

To describe our clustering model, let us assume that X is a data set containing D-dimensional binary vectors, i.e., $X \subset \{0,1\}^D$. Observe that in the case of sparse representation it is not profitable to remember the values at all positions since most of places are occupied by zero bits. Instead of saving the entire vector to the memory, it is more convenient to encode only the numbers of coordinates with bit 1 (see Figure 3). This strategy is commonly used in sparse vector structures.

Following the above motivation let us consider a distribution $p_1, \ldots, p_D$ of non-zero bits at particular coordinates. In other words, we ignore all empty bits and restrict the attention to the positions holding non-zero values. Therefore, the probability that j-th bit is non-zero equals:

$$p_j = \frac{P(x_j = 1)}{\sum_{l=1}^{D} P(x_l = 1)},$$

where $x = (x_1, \ldots, x_D)$ is a random vector representing a data set element.

The Shannon theory states that the code-lengths in an optimal prefix-free coding depend strictly on the associated probability distribution. Given a distribution $p_1, \ldots, p_D$ of positions with bit 1 it is possible to construct D codes, each with the length[1] $-\log p_j$. Observe that short codes correspond to the most frequent symbols, while the longest ones are related with rare objects. Given an arbitrary element

[1] in the limiting case

$x = (x_1, \ldots, x_D)$ we encode its non-zero coordinates and obtain that the memory cost equals

$$\sum_{j:x_j=1} -\log p_j$$

bits. The average code-length per one symbol is given by the Shannon entropy of $p_1, \ldots, p_D$:

$$\mathrm{h}(p_1, \ldots, p_D) = \sum_{j=1}^{D} p_j \cdot (-\log p_j).$$

Although we built prefix-free codes for non-zero coordinates, we have to be able to distinguish encoded representations of subsequent objects. It is realized by remembering the number of non-zero positions in a vector. Let q_t, for $t = 1, \ldots, D$, be a probability that exactly t positions contain non-zero bits:

$$q_t = P\left(\sum_{j=1}^{D} x_j = t\right).$$

We can also create prefix-free codes for a distribution $q_1, \ldots, q_D$. Therefore, if x contains t non-zero bits then its total code-length equals:

$$-\log q_t + \sum_{j:x_j=1} -\log p_j.$$

B. Clustering model

Real data sets are usually very complex and diverse. Since they are often generated by multiple sources, it might be inefficient to use a single encoder for the optimal description. It is more profitable to construct multiple encoders, each designed for one homogeneous part of data. Such an approach, which lies in the heart of our idea of clustering, leads to a natural division of a data set into groups.

As shown in the previous section the form of encoder depends strictly on the distribution of a data set. Since we aim at the minimization of statistical code-length, an optimal multi-encoder scheme should split data into groups, where each one contains instances described by the same simple probability distribution. Then, it would be possible to select specialized algorithm for each group which provides relatively short expected code-length. To obtain this goal, we partially follow MDLP and CEC approaches [5], [6].

Let us assume that $X_1, \ldots, X_k$ is a partition of X into pairwise disjoint sets. Every subset X_i is described by its own optimal coding algorithm. Observe that to encode an instance $x \in X_i$ one should remember a group identifier and the code of x defined by i-th encoder, i.e.,

$$\mathrm{code}(x) = [\mathrm{code}(X_i), \mathrm{code}_i(x)]. \tag{1}$$

Such a strategy enables the unique decoding because a retrieved coding algorithm allows subsequently for discovering an instance[2] (see also Figure 2). The compression procedure should find a division of X and design k-coding algorithms which minimize the expected length of code given by (1).

To design the optimal codes for i-th algorithm, we consider the probability distribution $p_1^i, \ldots, p_D^i$ of non-zero bits in X_i, i.e.,

$$p_j^i = \frac{P(x_j = 1 | x \in X_i)}{\sum_{j=1}^{D} P(x_j = 1 | x \in X_i)}$$

and the probabilities q_t^i, for $t = 1, \ldots, D$, that an arbitrary element of X_i contains exactly t non-zero bits:

$$q_t^i = P\left(\sum_{j=1}^{D} x_j = t | x \in X_i\right).$$

Let L_i denote the mean number of non-zero bits of a vector contained in X_i:

$$L_i = \sum_{t=1}^{D} t q_t^i. \tag{2}$$

Then, the average cost of encoding a vector by i-th algorithm is given by:

$$\begin{aligned} &\textstyle\sum_{t=1}^{D} q_t^i(-\log q_t^i) + L_i \cdot \sum_{j=1}^{D} p_j^i \cdot (-\log p_j^i) = \\ &\mathrm{h}(q_1^i, \ldots, q_D^i) + L_i \mathrm{h}(p_1^i, \ldots, p_D^i), \end{aligned} \tag{3}$$

To remember groups identifiers we optimize the code-lengths in a similar manner. Given a probability of generating an instance from a cluster X_i (the prior probability):

$$p^i = P(x \in X_i),$$

the optimal code-length of i-th identifier is given by

$$-\log p^i. \tag{4}$$

This term is a regularization factor and allows to keep the model as simple as possible. Since an introduction of any new cluster increases the total code-length, it might occur that maintaining less number of groups is more profitable. In other words, we penalize the model for its complexity.

The SEC clustering cost function gathers the formulas (3) and (4) and averages them over all clusters which represents the statistical cost of encoding a symbol in the multi-encoder sparse model.

SEC optimization problem. Let $X = \{0, 1\}^D$ be a data set of D-dimensional binary vectors. SEC aims at finding a partition of X into pairwise disjoint sets $X_1, \ldots, X_k$ which minimizes the average code-length using k encoders given by:

$$\sum_{i=1}^{k} p^i \cdot (-\log p^i + \mathrm{h}(q_1^i, \ldots, q_D^i) + L_i \mathrm{h}(p_1^i, \ldots, p_D^i)), \tag{5}$$

where L_i is defined by (2).

As mentioned, the regularization term in the SEC cost function allows for determining an optimal, from a compression point of view, number of clusters. In a basic data model generated from the mixture of two sources, we present an

[2]For a complete compression scheme we should also remember the header holding all codebooks. It can be discarded for very large data sets.

analytical criterion when it is more profitable to use two encoders instead of a single one. This result, even in its simplified form, is important, as it shows that (contrary to most clustering methods) in some cases it is profitable to reduce the number of groups.

Let us denote by $P(p, \alpha, d)$, for $p, \alpha \in [0, 1]$ and $d \in \{0, \ldots, D\}$, a distribution which generates bit 1 at j-th position with probability:

$$p_j = \begin{cases} p\alpha, & j = 1, \ldots, d, \\ p(1-\alpha), & j = d+1, \ldots, D. \end{cases} \tag{6}$$

Then our theoretical result is as follows:

Theorem 3.1: Let $X \subset \{0, 1\}^D$ be a data set generated from the mixture of two probability distributions:

$$\frac{1}{2}P(p, \alpha, d) + \frac{1}{2}P(p, 1-\alpha, d). \tag{7}$$

where $p, \alpha \in [0, 1]$ and $d = \frac{D}{2}$. Then it is more profitable to use two encoders determined by initial components instead of a single one, if and only if the following inequality holds:

$$-\alpha \log \alpha - (1-\alpha)\log(1-\alpha) < 1 - \frac{1}{pd}. \tag{8}$$

Before we formally prove the above statement, let us first give its interpretation. To visualize the situation we can arrange a data set in a matrix, where the rows correspond to the mixture components (instances), while the columns are related with their attributes:

$$\underbrace{\begin{pmatrix} p\alpha & p(1-\alpha) \\ p(1-\alpha) & p\alpha \end{pmatrix}}_{d \qquad d}$$

The matrix entries show the probability of generating bit 1 at a given coordinate belonging to one of four matrix regions.

The parameter α determines how similar are the instances generated from underlying distributions. For $\alpha = \frac{1}{2}$ both components are identical, while for $\alpha \in \{0, 1\}$ we get its perfect distinction. Observe that $L = pd$ is an average number of non-zero bits in a vector. Therefore, the decision of using one or two encoders (8) depends on the similarity level α and the average number of coordinates occupied by non-zero-bits. For instance, one can inspect that for small number of non-zero positions, e.g., $L = 5$ we use two encoders when $\alpha \in [0.25, 0.75]$, while for more dense data, e.g., $L = 50$ this interval is reduced to $[0.41, 0.59]$. Figure 4 illustrates this relation.

Proof: (of Theorem 3.1) To derive the inequality (8) we have to compare two cost functions for one and two clusters.

Let us observe that the probability that a vector contains t non-zero coordinates is identical in both models. Consequently, the average number of non-zero bits is also the same and equals:

$$pd\alpha + pd(1-\alpha) = pd.$$

The distribution of non-zero bits in the case of one cluster is given by:

$$\frac{\frac{1}{2}(p\alpha + p(1-\alpha))}{\frac{1}{2}2d(p\alpha + p(1-\alpha))} = \frac{1}{2d}.$$

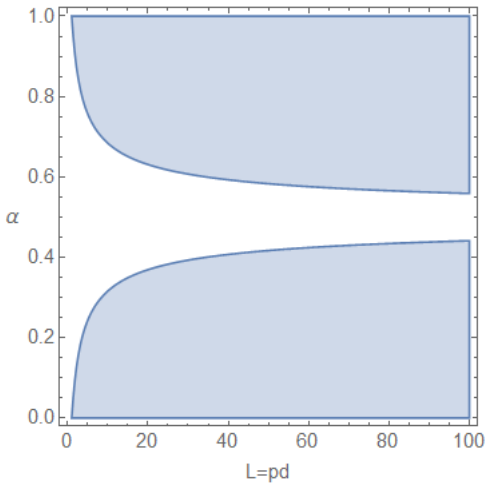

Fig. 4. **Illustration of Theorem 3.1.** Given a data set described by (7), it is more profitable to use two encoders instead of one in the blue area of graph. This region is determined by the relation between mixing level α and the mean number of non zero bits $L = pd$ in a vector.

On the other hand, the corresponding quantity for a distribution $P(p, \alpha, d)$ is given by:

$$\begin{cases} \frac{p\alpha}{pd\alpha + pd(1-\alpha)} & = \frac{\alpha}{d}, \quad \text{for } j = 1, \ldots, d, \\ \frac{p(1-\alpha)}{pd\alpha + pd(1-\alpha)} & = \frac{1-\alpha}{d}, \quad \text{for } j = d+1, \ldots, 2d. \end{cases}$$

The analogue formulas hold for a model generated from $P(p, a-\alpha, d)$.

Comparing the cost functions of underlying models, it is more efficient to use two encoders instead of a single one if:

$$(pd) \cdot 2d \cdot \tfrac{1}{2d}(-\log \tfrac{1}{2d}) > \\ 2 \cdot \tfrac{1}{2}\left(-\log \tfrac{1}{2} - (pd)d\tfrac{\alpha}{d}\log\tfrac{\alpha}{d} - (pd)d\tfrac{1-\alpha}{d}\log\tfrac{1-\alpha}{d}\right),$$

which is equivalent to

$$pd(1 + \log d) > 1 - pd\alpha \log \frac{\alpha}{d} - pd(1-\alpha)\log\frac{1-\alpha}{d}.$$

Elementary calculations lead to the inequality:

$$pd > 1 - pd(\alpha \log \alpha + (1-\alpha)\log(1-\alpha)),$$

which completes the proof. ∎

IV. Practical realization

In this section we show how to estimate the probabilities occurring in the SEC cost function (5) and present numerically efficient algorithm for its optimization.

We assume that a data set X is split into k clusters $X_1, \ldots, X_k$. Let us denote the number of elements in X_i with j-th position occupied by bit 1:

$$n_j^i = \sum_{x \in X_i} x_j, \text{ where } x = (x_1, \ldots, x_D) \in X$$

and a total number of non-zero positions within X_i:

$$s^i = \sum_{j=1}^{d} n_j^i.$$

1: **INPUT:**
2: $X \subset \{0,1\}^D$ - data set
3: $k \geq 2$ - initial number of clusters
4: **OUTPUT:**
5: Final partition $(X_1, \ldots, X_k)$ of X
6: **INITIALIZATION:**
7: $(X_1, \ldots, X_k) \leftarrow$ initial partition of X
8: **ITERATION:**
9: **while** NOT Done **do**
10: $Done \leftarrow True$
11: **for all** $x \in X$ **do**
12: $X_{new} \leftarrow \underset{Y \in (X_1, \ldots, X_k)}{argmax} \Delta E(x, Y)$
13: // membership minimizing the cost
14: **if** $X_{new} \neq x.cluster$ **then**
15: $Done \leftarrow False$
16: $Reassign(x, X_{new}, X_{old})$
17: // reassign x from X_{old} to X_{new}
18: $UpdateParams((X_{new}), (X_{old}), x)$
19: // recalculate clusters parameters
20: **end if**
21: **end for**
22: **end while**

Fig. 5. Pseudocode of SEC.

This allows to estimate the distribution of non-zero places in i-th group:

$$\hat{p}_j^i = \frac{n_j^i}{s^i}$$

and the probability that exactly t positions contain bit 1:

$$\hat{q}_t^i = \frac{|\{x \in X_i : \sum_{l=1}^d x_l = t\}|}{|X_i|}$$

Consequently, the mean amount of non-zero coordinates in X_i equals:

$$\hat{L}_i = \frac{s^i}{|X_i|}.$$

Finally, the prior probability of a group X_i can be approximated by the relative number of instances belonging to X_i:

$$\hat{p}^i = \frac{|X_i|}{|X|}.$$

After plugging these estimators into the SEC cost function, we get:

$$\log|X| + \frac{1}{|X|} \sum_i \left(\mathrm{h}(\hat{L}_1^i, \ldots, \hat{L}_d^i) + s^i \log s^i + \mathrm{h}(n_1^i, \ldots, n_d^i) \right). \tag{9}$$

To obtain an optimal division of X, the SEC cost function has to be minimized. Since it is not practically feasible to calculate its global minimum (see explanations given for k-means [24]), one can use some iterative algorithms to find one of its local minima. In the present paper we use a modified version of Hartigan procedure, which is commonly applied in an on-line version of k-means [7].

The minimization procedure consists of two steps: initialization and iteration. In the initialization stage, $k \geq 2$ nonempty groups are formed in an arbitrary manner. In the simplest case, it could be a random initialization, but to obtain better results one can also apply k-means++ seeding or use a partition returned by some fast and simple algorithm. In the iteration step the elements are reassigned between clusters in order to minimize the value of criterion function. A pseudocode of SEC is shown in Figure 5.

An efficient implementation of this algorithm requires fast recalculation of SEC cost function (9) after switching the element between clusters (line 18). Observe that a vector $n^i = (n_1^i, \ldots, n_D^i)$ can be updated after assigning a given element x to X_i by passing through its non-zero bits. This operation takes linear time with respect to the number of its non-zero coordinates for the sparse data structures. Similar argument holds for the recalculation of other clusters parameters, i.e., $s^i, |X_i|$ and corresponding entropies. In consequence, the computational complexity of one iteration of SEC (lines 11–21) can be approximated by the product of the total number of non-zero bits in the entire data set and the number of clusters.

V. Experimental results

In this section we demonstrate practical capabilities of SEC model and present a short evaluation study. The experiments were carried out on artificial examples, texts corpora and real data set of chemical compounds. We compared our technique with related methods commonly used in the case of binary data sets: k-medoids and hierarchical clustering with Jaccard distance function [3], and k-way method, the default algorithm implemented in Cluto software [15].

A. Artificial data sets

In the first experiment we examined the algorithms' sensitivity to data imbalance. For this purpose a data set $X \subset \{0,1\}^D$, where $D = 100$, was generated from the mixture of two binary sources:

$$\omega P(p, \alpha, d) + (1 - \omega) P(p, 1 - \alpha, d), \tag{10}$$

where $p = 0.1, \alpha = 0.05, d = D/2$ are fixed and ω changes from 0 to 1 (see (6) for the definition of distribution P). In other words, the probability of producing non-zero bit at any of d first positions equals $\alpha \cdot p = 0.005$ for the first component and $(1-\alpha) \cdot p = 0.095$ for the second. The converse situation holds for coordinates greater than d. Roughly speaking, the factor α controls the probability of generating bit 1 at the same position by both components – its low value makes the sources very different.

The mixing parameter ω specifies the number of instances produced by a particular probability distribution. One would expect that the clustering would reveal objects generated by particular component. Observe that for ω close to 0 or 1 the groups become extremely imbalanced which might be hard to discover by the algorithms. To verify the above hypothesis

[3] implemented in R package *cluster*

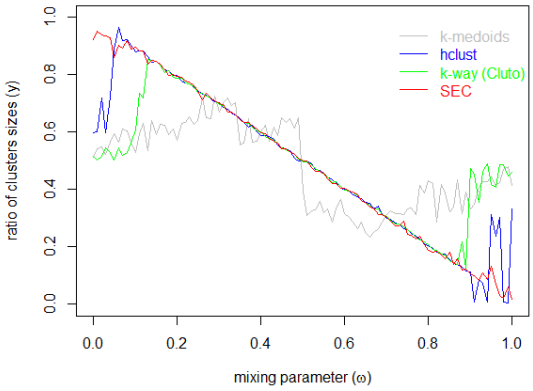

Fig. 6. **Imbalanced data.** The ratio of clusters sizes for a data set generated from the mixture of two binary sources (10). The number of instances produced by each component is controlled by the parameter ω. The optimal curve should be a linear function $y = 1 - \omega$.

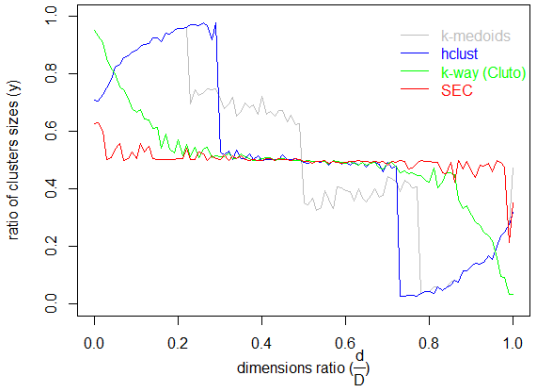

Fig. 7. **Imbalanced features.** The ratio of clusters sizes for data generated from the mixture of two binary sources (11). The number of features characteristic for each component is controlled by the parameter d. The optimal curve should be a constant function $y = \frac{1}{2}$.

we ran each method with two groups and reported resulting clusters sizes, Figure 6. The optimal curve preserving the original components equals $y = 1 - \omega$.

It is evident that k-medoids is not completely able to detect the natural structure of data. Other algorithms gave satisfactory results for $\omega \in [0.1, 0.9]$. Out of this range k-way began to create groups of similar sizes. Slightly better results were produced by hierarchical clustering method, but the negative effect caused by the data imbalance appeared for $\omega < 0.05$ and $\omega > 0.95$. The SEC was the most stable and robust on the change of parameter ω.

In the second experiment we considered a data set sampled from the mixture of sources given by:

$$\frac{1}{2}P(p, \alpha, d) + \frac{1}{2}P(p, 1 - \alpha, D - d), \tag{11}$$

where $p = 0.1, \alpha = 0.05, D = 100$ are constants and d ranges form 0 to D. When $d < \frac{D}{2}$ then the second source is identified by the smaller number of bits than the first one. Therefore, by changing the value of parameter d we scale the number of features characteristic for components.

The optimal clustering should be invariant with respect to the shift of dimension bound d. Since the number of instances produced by every distribution is constant then the clusters should remain equally-sized. Each algorithm was run with two clusters and the ratio of returned clusters sizes was marked in the Figure 7. The full invariance corresponds to the constant curve $y = \frac{1}{2}$.

The results produced by SEC and k-way for $\frac{d}{D} \in [0.2, 0.8]$ are very similar. Nevertheless, outside of this range k-way began gradually assigning more objects to one cluster. This negative effect was not so evident in the case of SEC, which preserved the natural proportions of clusters sizes. Both baseline methods gave very inaccurate results which confirms that they are very sensitive to such data structure.

B. Texts corpora

Set-of-words is the simplest vector representation a text. Given a dictionary of words, a sentence is represented as a binary vector, where coordinates indicate the presence or absence of words from a dictionary in a sentence. We utilized set-of-words representation in tests including four text data sets which are summarized in Table I (first four rows) [25], [26].

As an external clustering criterion the Adjusted Rand Index (ARI) was utilized which is the well-known measure of compatibility of a constructed grouping with a reference partition [27]. For a perfect clustering (partition identical with a reference grouping) ARI equals 1, while for a random partition ARI takes value 0. To conduct statistical tests we ran each clustering method 10 times on 60% of randomly selected instances from each data set. In Figure 8 we reported the mean ARI values calculated over all 10 runs.

One can observe that for *questions* and *sentiment* sets no algorithm was able to construct a partition close to the reference one. It could be caused by an extreme sparsity of data, where only 4 and 7 coordinates, respectively, were non-zero on average. On the other hand, SEC outperformed all algorithms on the remaining two sets. For *20newsgroups* it achieved very high ARI (close to 0.6), while for *farm-ads* it was the only method which was able to produce partition a little bit similar to the reference one. The Wicoxon signed ranked test proved that at 0.01 significance level there was no evidence to reject the hypothesis that SEC gives higher ARI than other algorithms.

In NLP the perplexity is one of the most popular internal measures for comparing clustering algorithms [28]. It is determined by the entropy of n-gram language model constructed on a given set, for $n \geq 2$. Since the frequency of bigrams or trigrams usually is independent from set-of-words representation, it will not be reasonable to apply such a measure for comparing obtained results. Therefore, the goodness of a partition was measured by calculating the average entropy per word within clusters $X_1, \ldots, X_k$, i.e.,

$$p(X_1)\mathrm{h}(p_1^1, \ldots, p_D^1) + \ldots + p(X_k)\mathrm{h}(p_1^k, \ldots, p_D^k),$$

TABLE I
SUMMARY OF DATA SETS USED IN EXPERIMENTS.

Data set	Size	Dimensions	Avg. number of non-zero bits	Classes
20newsgroups	6997	26411	99.49	7
farm-ads	4143	54877	197.23	2
questions	5452	3029	4.04	6
sentiment	1000	2750	7.50	2
Klekota-Roth	3696	4860	64.28	28
Extended	3696	1024	366.97	28

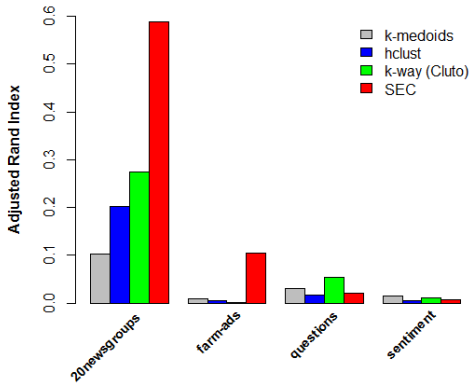

Fig. 8. Adjusted Rand Index for a texts data clustering.

where $p(X_i)$ and p_j^i are defined in Section IV. A partition was considered to be better if it had lower entropy value.

The results illustrated in Figure 9 indicate that SEC gave the lowest entropy value for all sets in a statistically significant way. This is probably caused by the fact that SEC partially uses entropy in its clustering criterion function. The worst results were produced by hierarchical clustering, while the performance of k-way was slightly better than k-medoids.

C. Chemical compounds

Chemical compounds are usually represented as fingerprints, i.e., high dimensional bit sequences which encode the presence or absence of various chemical patterns. Since different chemical features can be taken into account there were constructed many kinds of fingerprints. Their lengths range from 79 bits (Estate FP) to 4860 bits (Klekota-Roth FP) [29]. This representation is similar to set-of-words employed in text analysis.

In the experiment we considered a real data set containing compounds which are active with respect to 5-HT_{1A} receptor – one of the proteins responsible for regulation of central nervous system. This data set was manually clustered by the expert into 28 chemical classes based on their structural features [30]. Two fingerprints were utilized for compounds representations: Klekota-Roth FP and Extended FP which were proven to be the most informative ones (see last two rows of Table I) [31].

To evaluate the algorithms we measured the similarities of constructed partitions with aforementioned reference grouping

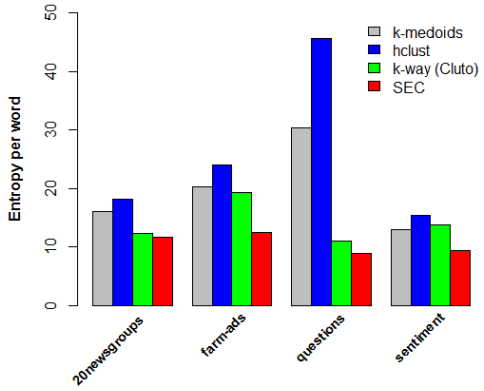

Fig. 9. Average entropy per word within clusters for texts data clustering.

quantified by ARI. Making use of the analogy between set-of-words and fingerprint representations, we also calculated the average entropy per feature within clusters. All algorithms were run 10 times on 60% randomly selected instances with 28 clusters and obtained average ARI and entropy values were reported in Figures 10 and 11.

One can observe that the difference between algorithms performances is slight. Nevertheless, SEC and k-way gave the highest ARI for both fingerprints on an average. The compatibility of the partition constructed by k-way with reference grouping was higher in the case of Klekota-Roth FP, while SEC returned more accurate results in the case of Extended FP. At 0.01 significance level there was no statistical evidence to reject these hypotheses.

On the other hand, SEC gave the lowest entropy for Klekota-Roth, while the k-medoids was better for Extended FP. This behavior of SEC might be explained by the fact that in some sense it tries to find a true probability distribution of data assuming its sparse representation. The average number of non-zero bits for a compound representing by Klekota-Roth FP equals 64 out of 4860 bits and 366 out of 1024 for Extended FP. Therefore, the sparse representation is more suitable in the case of Klekota-Roth.

VI. CONCLUSION

In this paper we introduced SEC model for clustering of sparse high dimensional binary data. The strength of the method is that:

- It aims at finding true probability distribution of data assuming its sparse representation.
- It follows the MDLP approach and therefore can be implemented as a compression algorithm.
- The model is able to select the optimal number of groups by adding the cost of clusters identification.
- Its criterion function can be optimized with use the iterative Hartigan algorithm which allows for fast reassigning the elements between clusters.
- The complexity of a single iteration of the algorithm is determined by a total number of non-zero coordi-

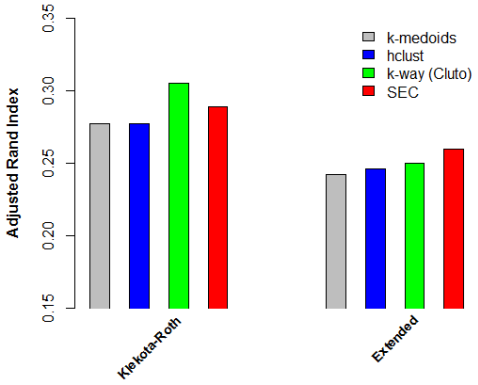

Fig. 10. Adjusted Rand Index for a chemical data clustering.

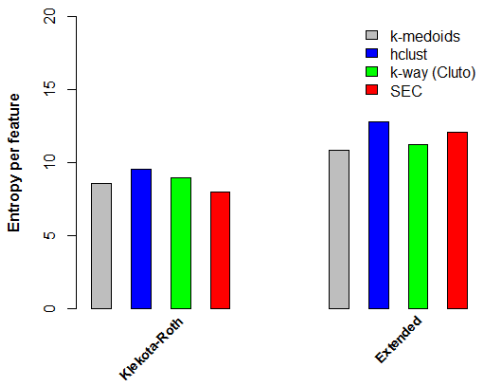

Fig. 11. Average entropy per feature within clusters for a chemical data clustering.

nates, while the influence of the number of instances is marginal.

ACKNOWLEDGMENT

The work of the first author was partially supported by the National Centre of Science (Poland) grant no. 2014/13/N/ST6/01832, while the third author was supported by the National Centre of Science (Poland) grant no. 2014/13/B/ST6/01792.

REFERENCES

[1] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009, vol. 344.

[2] E. Bielska, X. Lucas, A. Czerwoniec, J. M. Kasprzak, K. H. Kaminska, and J. M. Bujnicki, "Virtual screening strategies in drug design–methods and applications," *BioTechnologia. Journal of Biotechnology Computational Biology and Bionanotechnology*, vol. 92, no. 3, 2011.

[3] B. Roark, M. Saraclar, and M. Collins, "Discriminative n-gram language modeling," *Computer Speech & Language*, vol. 21, no. 2, pp. 373–392, 2007.

[4] M. Steinbach, L. Ertöz, and V. Kumar, "The challenges of clustering high dimensional data," in *New Directions in Statistical Physics*. Springer, 2004, pp. 273–309.

[5] J. Rissanen, "Minimum-description-length principle," *Encyclopedia of statistical sciences*, 1985.

[6] J. Tabor and P. Spurek, "Cross-entropy clustering," *Pattern Recognition*, vol. 47, no. 9, pp. 3046–3059, 2014.

[7] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Applied statistics*, pp. 100–108, 1979.

[8] H. Bock, "Probabilistic aspects in cluster analysis," in *Conceptual and numerical analysis of data*. Springer, 1989, pp. 12–44.

[9] G. Celeux and G. Govaert, "Clustering criteria for discrete data and latent class models," *Journal of classification*, vol. 8, no. 2, pp. 157–176, 1991.

[10] T. Li, S. Ma, and M. Ogihara, "Entropy-based criterion in categorical clustering," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 68.

[11] H. Li, K. Zhang, and T. Jiang, "Minimum entropy clustering and applications to gene expression analysis," in *Computational Systems Bioinformatics Conference, 2004. CSB 2004. Proceedings. 2004 IEEE*. IEEE, 2004, pp. 142–151.

[12] T. Li, "A unified view on clustering binary data," *Machine Learning*, vol. 62, no. 3, pp. 199–215, 2006.

[13] ——, "A general model for clustering binary data," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 2005, pp. 188–197.

[14] D. Barbará, Y. Li, and J. Couto, "Coolcat: an entropy-based algorithm for categorical clustering," in *Proceedings of the eleventh international conference on Information and knowledge management*. ACM, 2002, pp. 582–589.

[15] G. Karypis, "Cluto-a clustering toolkit," DTIC Document, Tech. Rep., 2002.

[16] J. C. Xavier, A. M. Canuto, N. D. Almeida, and L. M. Goncalves, "A comparative analysis of dissimilarity measures for clustering categorical data," in *Neural Networks (IJCNN), The 2013 International Joint Conference on*. IEEE, 2013, pp. 1–8.

[17] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[18] R. Mall, S. Mehrkanoon, R. Langone, J. Suykens *et al.*, "Optimal reduced sets for sparse kernel spectral clustering," in *Neural Networks (IJCNN), 2014 International Joint Conference on*. IEEE, 2014, pp. 2436–2443.

[19] S. Wang, F. Chen, and J. Fang, "Spectral clustering of high-dimensional data via nonnegative matrix factorization," in *Neural Networks (IJCNN), 2015 International Joint Conference on*. IEEE, 2015, pp. 1–8.

[20] L. Ma, C. Wang, and B. Xiao, "Sparse representation based on matrix rank minimization and k-means clustering for recognition," in *Neural Networks (IJCNN), The 2012 International Joint Conference on*. IEEE, 2012, pp. 1–8.

[21] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for k-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336–3341, 2009.

[22] V. Batagelj, "Generalized ward and related clustering problems," *Classification and related methods of data analysis*, pp. 67–74, 1988.

[23] M. Smieja and J. Tabor, "Spherical wards clustering and generalized voronoi diagrams," in *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on*. IEEE, 2015, pp. 1–10.

[24] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[25] D. N. A. Asuncion, "UCI machine learning repository," 2007. [Online]. Available: http://www.ics.uci.edu/$\sim$mlearn/{MLR}epository.html

[26] X. Li and D. Roth, "Learning question classifiers," in *Proceedings of the 19th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 2002, pp. 1–7.

[27] L. Hubert and P. Arabie, "Comparing partitions," vol. 2, no. 1, pp. 193–218, 1985.

[28] L. Rigouste, O. Cappé, and F. Yvon, "Evaluation of a probabilistic method for unsupervised text clustering," in *Proceedings of the international symposium on applied stochastic models and data analysis (ASMDA), Brest, France*, 2005.

[29] S. Riniker and G. A. Landrum, "Open-source platform to benchmark fingerprints for ligand-based virtual screening," *Journal of cheminformatics*, vol. 5, no. 1, pp. 1–17, 2013.

[30] D. Warszycki, S. Mordalski, K. Kristiansen, R. Kafel, I. Sylte, Z. Chilmonczyk, and A. J. Bojarski, "A linear combination of pharmacophore hypotheses as a new tool in search of new active compounds - an application for 5-ht$_{1A}$ receptor ligands," vol. 8, no. 12, p. e84510, 12 2013.

[31] M. Śmieja, D. Warszycki, J. Tabor, and A. J. Bojarski, "Asymmetric clustering index in a case study of 5-ht1a receptor ligands," *PLoS ONE*, vol. 9(7): e102069. doi: 10.1371/journal.pone.0102069, 2014.